

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

ATTORNEY DOCKET NO.:  
**PAVT016US0**

TITLE:  
**A SYSTEM AND METHOD OF ADAPTIVE CONTROL OF PROCESSES WITH  
VARYING DYNAMICS**

INVENTOR(S):  
**Bijan Sayyarrodsari, Eric Hartman, Celso Axelrud  
and Kidir Liano**

SUBMITTED BY:

Hulsey, Grether, Fortkort & Webster, LLP  
8911 N. Capital of Texas Hwy., Suite 3200  
Austin, Texas 78759  
(512) 795-0095 - Telephone  
(512) 795-9905 - Facsimile

CERTIFICATE OF EXPRESS MAILING UNDER 37 CFR § 1.10

I hereby certify that this correspondence is being deposited with the United States Postal Service Express Mail Service under 37 C.F.R. §1.10 addressed to: Mail Stop Provisional Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22213-1450, on this the 9 day of December, 2003.

Express Mailing Number: EV 417743502 US

  
Mary Schnaiter

**A SYSTEM AND METHOD OF ADAPTIVE CONTROL  
OF PROCESSES WITH VARYING DYNAMICS**

TECHNICAL FIELD OF THE INVENTION

**[0001]** The present invention relates generally to advanced predictive modeling and control. More particularly the invention relates to adaptive control which is particularly useful for modeling and control of processes with varying dynamics characteristics.

BACKGROUND OF THE INVENTION

**[0002]** Many operating processes have varying dynamics characteristics which are notoriously difficult to model and control. These processes are extremely diverse and can be found in virtually any field of endeavor. One example of such operating processes is particle accelerators used to study fundamental particles. The study of fundamental particles and their interactions seeks to answer two questions: (1) what are the fundamental building blocks (smallest) from which all matter is made; and (2) what are the interactions between these particles that govern how the particles combine and decay? To answer these questions, physicist use accelerators to provide high energy to subatomic particles, which then collide with targets. Out of these interactions come many other subatomic particles that pass into detectors. FIGURES 1A and 1B illustrate typical collisions or interactions used in this

1 study. From the information gathered in the detectors,  
2 physicists can determine properties of the particles and their  
3 interactions.

4 **[0003]** In these experiments, subatomic particles collide.  
5 However, to achieve the desired experiments requires a large  
6 degree of control over the particles trajectory and the  
7 environment in which the collisions actually take place.  
8 Process and control models are typically used to aid the  
9 physicist in the setup and execution of these experiments.

10 **[0004]** Process Models used for prediction, control, and  
11 optimization can be divided into two general categories, steady  
12 state models and dynamic models. These models are mathematical  
13 constructs that characterize the process, and process  
14 measurements are often utilized to build these mathematical  
15 constructs in a way that the model replicates the behavior of  
16 the process. These models can then be used for prediction,  
17 optimization, and control of the process.

18 **[0005]** Many modern process control systems use steady-state  
19 or static models. These models often capture the information  
20 contained in large amounts of data, wherein this data typically  
21 contains steady-state information at many different operating  
22 conditions. In general, the steady-state model is a non-linear  
23 model wherein the process input variables are represented by the  
24 vector U that is processed through the model to output the

1 dependent variable Y. The non-linear steady-state model is a  
2 phenomenological or empirical model that is developed utilizing  
3 several ordered pairs  $(U_i, Y_i)$  of data from different measured  
4 steady states. If a model is represented as:

$$Y=P(U, Y) \quad (1)$$

6 where P is an appropriate static mapping, then the steady-  
7 state modeling procedure can be presented as:

$$M(\bar{U}, \bar{Y}) \rightarrow P \quad (2)$$

9 where U and Y are vectors containing the  $U_i, Y_i$  ordered pair  
10 elements. Given the model P, then the steady-state process  
11 gain can be calculated as:

$$K = \frac{\Delta P(u, y)}{\Delta u} \quad (3)$$

13 The steady-state model, therefore, represents the process  
14 measurements taken when the process is in a "static" mode. These  
15 measurements do not account for process behavior under non-  
16 steady-state condition (e.g. when the process is perturbed, or  
17 when process transitions from one steady-state condition to  
18 another steady-state condition). It should be noted that real  
19 world processes (e.g. particle accelerators, chemical plants)  
20 operate within an inherently dynamic environment. Hence steady-  
21 state models alone are, in general, not sufficient for  
22 prediction, optimization, and control of an inherently dynamic  
23 process.

1   **[0006]**     A dynamic model is typically a model obtained from  
2   non-steady-state process measurements. These non-steady-state  
3   process measurements are often obtained as the process  
4   transitions from one steady-state condition to another. In this  
5   procedure, process inputs (manipulated and/or disturbance  
6   variables denoted by vector  $u(t)$ ), applied to a process affect  
7   process outputs (controlled variables denoted by vector  $y(t)$ ),  
8   that are being output and measured. Again, ordered pairs of  
9   measured data ( $u(t_1)$ ,  $y(t_1)$ ) represent a phenomenological or  
10   empirical model, wherein in this instance data comes from non-  
11   steady-state operation. The dynamic model is represented as:

$$y(t) = p(u(t), u(t-1), \dots, u(t-M), y(t), y(t-1), \dots, y(t-N)) \quad (4)$$

13     where  $p$  is an appropriate mapping.  $M$  and  $N$  specify the  
14     input and output history that is required to build the  
15     dynamic model.

16   The state-space description of a dynamic system is equivalent to  
17   input/output description in Equation (4) for appropriately  
18   chosen  $M$  and  $N$  values, and hence the description in Equation (4)  
19   encompasses state-space description of the dynamic  
20   systems/processes as well.

21   **[0007]**     Nonlinear dynamic systems are in general difficult to  
22   build. Prior art includes a variety of model structures in which  
23   a nonlinear static model and a linear dynamic model are combined  
24   in order to represent a nonlinear dynamic system. Examples

1 include Hammerstein models (where a static nonlinear model  
2 precedes a linear dynamic model in a series connection), and  
3 Wiener models (where a linear dynamic model precedes a static  
4 nonlinear model in a series connection). Patent #5,933,345  
5 constructs a nonlinear dynamic model in which the nonlinear  
6 model respects the nonlinear static mapping captured by a neural  
7 network.

8 **[0008]** This invention extends the state of the art by  
9 developing a neural network that is trained to produce the  
10 variation in parameters of a dynamic model that can best  
11 approximate the dynamic mapping in Equation (4), and then  
12 utilizing the overall input/output static mapping (also captured  
13 with a neural network trained according to the description in  
14 paragraph [0005]) to construct a parsimonious nonlinear dynamic  
15 model appropriate for prediction, optimization, and control of  
16 the process it models.

17 **[0009]** In most real-world applications, first-principles  
18 (FPs) models (FPMs) describe (fully or partially) the laws  
19 governing the behavior of the process. Often, certain  
20 parameters in the model critically affect the way that model  
21 behaves. Hence, the design of a successful control system  
22 depends heavily on the accuracy of the identified parameters.  
23 This invention develops a parametric structure for the nonlinear  
24 dynamic model that represents the process (see Equation (6)). To

1 fulfill online modeling system goals, neural networks (NNs)  
2 models (NNMs) have been developed to robustly identify the  
3 variation in the parameters of this dynamic model, when the  
4 operation region changes considerably (see Figure 7). The  
5 training methodology developed can also be used to robustly  
6 train parametric steady-state models.

7 **[0010]** Numerous ways of combining NNMs and FPMs exist. NNMs  
8 and FPMs can be combined "in parallel". Here the NNMs the  
9 errors of the FPMs, then add the outputs of the NNM and the FPM  
10 together. This invention uses a combination of the empirical  
11 model and parametric physical models in order to model a  
12 nonlinear process with varying dynamics.

13 **[0011]** NNMs and FPMs represent two different methods of  
14 mathematical modeling. NNMs are empirical methods for doing  
15 nonlinear (or linear) regression (i.e., fitting a model to  
16 data). FPMs are physical models based on known physical  
17 relationships. The line between these two methods is not  
18 absolute. For example, FPMs virtually always have "parameters"  
19 which must be fit to data. In many FPMs, these parameters are  
20 not in reality constants, but vary across the range of the  
21 model's possible operation. If a single point of operation is  
22 selected and the model's parameters are fitted at that point,  
23 then the model's accuracy degrades as the model is used farther  
24 and farther away from that point. Sometimes multiple FPMs are

1 fitted at a number of different points, and the model closest to  
2 the current operating point is used as the current model.

3 **[0012]** NNMs and FPMs each have their own set of strengths and  
4 weaknesses. NNMs typically are more accurate near a single  
5 operating point while FPMs provide better extrapolation results  
6 when used at an operating point distant from where the model's  
7 parameters were fitted. This is because NNMs contain the  
8 idiosyncrasies of the process being modeled. These sets of  
9 strengths and weaknesses are highly complementary - where one  
10 method is weak the other is strong - and hence, combining the  
11 two methods can yield models that are superior in all aspects to  
12 either method alone. This is applicable to the control of  
13 processes where dynamic behavior of the process displays  
14 significant variations over the operation range of the process.

15 **[0013]** The present invention provides an innovative approach  
16 to building parametric nonlinear models that are computationally  
17 efficient representations of both steady-state and dynamic  
18 behavior of a process over its entire operation region. For  
19 example, the present invention provides a system and method for  
20 controlling nonlinear control problems within particle  
21 accelerators. This method involves first utilizing software  
22 tools to identify input variables and controlled variables  
23 associated with the operating process to be controlled, wherein  
24 at least one input variable is a manipulated variable. This



1 software tool is further operable to determine relationships  
2 between the input variables and controlled variables. A control  
3 system that provides inputs to and acts on inputs from the  
4 software tools tunes one or more model parameters to ensure a  
5 desired behavior for one or more controlled variables, which in  
6 the case of a particle accelerator may be realized as a more  
7 efficient collision.

8 **[0014]** The present invention may determine relationships  
9 between input variables and controlled variables based on a  
10 combination of physical models and empirical data. This  
11 invention uses the information from physical models to robustly  
12 construct the parameter varying model of Figure 7 in a variety  
13 of ways that includes but is not limited to generating data from  
14 the physical models, using physical models as constraints in  
15 training of the neural networks, and analytically approximating  
16 the physical model with a model of the type described in  
17 Equation (6).

18 **[0015]** The parametric nonlinear model of Figure (7) can be  
19 augmented with a parallel, neural networks that models the  
20 residual error of the series model. The parallel neural network  
21 can be trained in a variety of ways that includes concurrent  
22 training with the series neural network model, independent  
23 training from the series neural networks model, or iterative  
24 training procedure.

1   **[0016]**     The neural networks utilized in this case may be  
2   trained according to any number of known methods. These methods  
3   include both gradient-based methods, such as back propagation  
4   and gradient-based nonlinear programming (NLP) solvers (for  
5   example sequential quadratic programming, generalized reduced  
6   gradient methods), and non-gradient methods. Gradient-based  
7   methods typically require gradients of an error with respect to  
8   a weight and bias obtained by either numerical derivatives or  
9   analytical derivatives.

10   **[0017]**     In the application of the present invention to a  
11   particle accelerator, controlled variables such as but not  
12   limited to varying magnetic field strength, shape, location  
13   and/or orientation are controlled by adjusting corrector magnets  
14   and/or quadrapole magnets to manipulate particle beam positions  
15   within the accelerator so as to achieve more efficient  
16   interactions between particles.

17   **[0018]**     Another embodiment of the present invention takes the  
18   form of a system for controlling nonlinear control problems  
19   within particle accelerators. This system includes a  
20   distributed control system used to operate the particle  
21   accelerator. The distributed control system further includes  
22   computing device(s) operable to execute a first software tool  
23   that identifies input variables and controlled variables  
24   associated with the given control problem in particle

1 accelerator, wherein at least one input variable is a  
2 manipulated variable. The software tool is further operable to  
3 determine relationships between the input variables and  
4 controlled variables. Input/output controllers (IOCs) operate  
5 to monitor input variables and tune the previously identified  
6 control variable(s) to achieve a desired behavior in the  
7 controlled variable(s).

8 **[0019]** The physical model in Figure 7 is shown as a function  
9 of the input variables. It is implied that if variation of a  
10 parameter in the dynamic model is a function of one or more  
11 output variables of the process, then the said output variables  
12 are treated as inputs to the neural-network model. The  
13 relationship between the input variables and the parameters in  
14 the parametric model may be expressed through the use of  
15 empirical methods, such as but not limited to neural networks.

16 **[0020]** Specific embodiments of the present invention may  
17 utilize IOCs associated with corrector magnets and/or quadruple  
18 magnets to control magnetic field strength, shape, location  
19 and/or orientation and in order to achieve a desired particle  
20 trajectory or interaction within the particle accelerator.

21 **[0021]** Yet another embodiment of the present invention  
22 provides a dynamic controller for controlling the operation of a  
23 particle accelerator by predicting a change in the dynamic input  
24 values to effect a change in the output of the particle

1 accelerator from a current output value at a first time to a  
2 different and desired output value at a second time in order to  
3 achieve more efficient collisions between particles. This  
4 dynamic controller includes a dynamic predictive model for  
5 receiving the current input value, wherein the dynamic  
6 predictive model changes dependent upon the input value, and the  
7 desired output value. This allows the dynamic predictive model  
8 to produce desired controlled input values at different time  
9 positions between the first time and the second time so as to  
10 define a dynamic operation path of the particle accelerator  
11 between the current output value and the desired output value at  
12 the second time. An optimizer optimizes the operation of the  
13 dynamic controller over the different time positions from the  
14 first time to the second time in accordance with a predetermined  
15 optimization method that optimizes the objectives of the dynamic  
16 controller to achieve a desired path from the first time to the  
17 second time, such that the objectives of the dynamic predictive  
18 model from the first time to the second time vary as a function  
19 of time.

20 **[0022]** A dynamic forward model operates to receive input  
21 values at each of time positions and maps the input values to  
22 components of the dynamic predictive model associated with the  
23 received input values in order to provide a predicted dynamic  
24 output value. An error generator compares the predicted dynamic

1 output value to the desired output value and generates a primary  
2 error value as the difference for each of the time positions.  
3 An error minimization device determines a change in the input  
4 value to minimize the primary error value output by the error  
5 generator. A summation device for summing said determined input  
6 change value with an original input value, which original input  
7 value comprises the input value before the determined change  
8 therein, for each time position to provide a future input value  
9 as a summed input value. A controller operates the error  
10 minimization device to operate under control of the optimizer to  
11 minimize said primary error value in accordance with the  
12 predetermined optimization method.

1                    BRIEF DESCRIPTION OF THE DRAWINGS

2    **[0023]**      For a more complete understanding of the present  
3    invention and the advantages thereof, reference is now made to  
4    the following description taken in conjunction with the  
5    accompanying drawings in which like reference numerals indicate  
6    like features and wherein:  
7    FIGURES 1A and 1B illustrate typical collisions or interactions  
8    studied with particle accelerators;  
9    FIGURE 2 depicts the components of a particle accelerator  
10    operated and controlled according to the system and method of  
11    the present invention;  
12    FIGURE 3 illustrates a polarized electron gun associated with a  
13    particle accelerator operated and controlled according to the  
14    system and method of the present invention;  
15    FIGURE 4 depicts a multi-layer detector associated with a  
16    particle accelerator operated and controlled according to the  
17    system and method of the present invention;  
18    FIGURE 5 depicts the three physical layers associated with a  
19    particle accelerator operated and controlled according to the  
20    system and method of the present invention;  
21    FIGURE 6 depicts the five software layers associated with a  
22    particle accelerator operated and controlled according to the  
23    system and method of the present invention;

1 FIGURE 7 illustrates the interaction between a neural network  
2 model and a parametric dynamic or static model;  
3 FIGURE 8 provides a screenshot that evidences the clear  
4 correlation between the MVs with the BPM;  
5 FIGURE 9 provides yet another screenshot of the variation in  
6 variables; and  
7 FIGURE 10 provides yet another screen shot showing a capture of  
8 the input/output data.  
9 FIGURE 11 displays one such input/output relationship for the  
10 SPEAR Equipment at SLAC.  
11 FIGURE 12 illustrates the relationship of the various models in  
12 the controller and the controller and the process.

DETAILED DESCRIPTION OF THE INVENTION

**[0024]** Preferred embodiments of the present invention are illustrated in the FIGUREs, like numerals being used to refer to like and corresponding parts of the various drawings.

**[0025]** The present invention provides methodologies for the computationally efficient modeling of processes with varying dynamics. More specifically, the present invention provides a method for robust implementation of indirect adaptive control techniques in problems with varying dynamics through transparent adaptation of the parameters of the process model that is used for prediction and online optimization. Such problems include but are not limited to the control of: particle trajectories within particle accelerators, temperature in a chemical reactors, and grade transition in a polymer manufacturing process.

**[0026]** This innovation enables improvement of existing control software, such as Pavilion Technology's Process Perfecter®, to exert effective control in problems with even severely varying dynamics. This is especially well suited for the control of particle trajectories within accelerators.

**[0027]** The parametric nonlinear model introduced in this invention has been successfully used by inventors to model severely nonlinear processes. One specific application directly



1 relates to the control of the linear accelerator at Stanford  
2 Linear Accelerator Center (SLAC).

3 **[0028]** The present invention provides a powerful tool for the  
4 analysis of the nonlinear relationship between the  
5 manipulated/disturbance variables and the controlled variables  
6 such as those at the Stanford Position Electron Asymmetric Ring  
7 (SPEAR). Tuning of the control variables can benefit from this  
8 analysis. SLAC performs and supports world-class research in  
9 high-energy physics, particle astrophysics and disciplines using  
10 synchrotron radiation. To achieve this it is necessary to  
11 provide accelerators, detectors, instrumentation, and support  
12 for national and international research programs in particle  
13 physics and scientific disciplines that use synchrotron  
14 radiation. The present invention plays a key role in advances  
15 within the art of accelerators, and accelerator-related  
16 technologies and devices specifically and generally to all  
17 advanced modeling and control of operating processes -  
18 particularly those that exhibit severe nonlinear behavior that  
19 vary over time.

20 **[0029]** Accelerators such as those at SLAC provide high energy  
21 to subatomic particles, which then collide with targets. Out of  
22 these interactions come many other subatomic particles that pass  
23 into detectors. From the information gathered in the detector,

1 physicists determine properties of the particles and their  
2 interactions.

3 **[0030]** The higher the energy of the accelerated particles,  
4 the more fully the structure of matter may be understood. For  
5 that reason a major goal is to produce higher and higher  
6 particle energies. Hence, improved control systems are required  
7 to ensure the particles strike their targets as designed within  
8 the experiment.

9 **[0031]** Particle accelerators come in two designs, linear and  
10 circular (synchrotron). The accelerator at SLAC is a linear  
11 accelerator. The longer a linear accelerator is, the higher the  
12 energy of the particles it can produce. A synchrotron achieves  
13 high energy by circulating particles many times before they hit  
14 their targets.

15 **[0032]** The components of a particle accelerator 10 are  
16 illustrated in FIGURE 2. At the leftmost end of FIGURE 2 is  
17 electron gun 12, which produces the electrons 14 to be  
18 accelerated. Any filament that is heated by an electrical  
19 current flowing through the filament releases electrons.  
20 Electric field 16 then accelerates electrons 14 towards the  
21 beginning of accelerator 18.

22 **[0033]** Alternatively, a polarized electron gun 20, as shown  
23 in FIGURE 3, may be used. Here polarized laser light from laser  
24 sources 22 knocks electrons 24 off the surface of semiconductor

1 26. Electric field 30 then accelerates the electrons toward  
2 accelerator pipe 32. Polarized electron gun 20 must be kept at  
3 an extremely high vacuum, even higher than that of the  
4 accelerator itself. Such a vacuum may be on the order of  $10^{-12}$   
5 Tor.

6 **[0034]** Returning to FIGURE 2, after the first few feet of the  
7 linear accelerator 18, the electrons 14 are traveling in bunches  
8 with an energy of approximately 10 MeV<sup>G</sup>. This means that  
9 electrons 14 have reached 99.9% the speed of light. These  
10 bunches of electrons 14 have a tendency to spread out in the  
11 directions perpendicular to their travel.

12 **[0035]** Because a spread-out beam gives fewer collisions than  
13 a narrowly focused one, the electron and positron bunches are  
14 sent into damping rings 33 (electrons to north, positrons to  
15 south). These are small storage rings located on either side of  
16 the main accelerator. As the bunches circulate in damping rings  
17 33, electrons 14 lose energy by synchrotron radiation and are  
18 reaccelerated each time they pass through a cavity fed with  
19 electric and magnetic fields. The synchrotron radiation  
20 decreases the motion in any direction, while the cavity  
21 reaccelerates only those in the desired direction. Thus, the  
22 bunch of electrons or positrons becomes increasingly parallel in  
23 motion as the radiation "damps out" motion in the unwanted  
24 directions. The bunches are then returned to accelerator 18 to

1 gain more energy as travel within it. Further focusing is  
2 achieved with a quadrapole magnet or connector magnet 16 in  
3 beamlines. Focusing here is achieved in one plane while  
4 defocusing occurs in the other.

5 **[0036]** Bunches of electrons 14 are accelerated within  
6 accelerator 18 in much the same way a surfer is pushed along a  
7 wave. The electromagnetic waves that push the electrons in  
8 accelerator 18 are created by high-energy microwaves. These  
9 microwaves emit from klystrons (not shown) and feed into the  
10 particle accelerator structure via waveguides to create a  
11 pattern of electric and magnetic fields.

12 **[0037]** Inside accelerator 18, the microwaves from the  
13 klystrons set up currents that cause oscillating electric fields  
14 pointing along accelerator 18 as well as oscillating magnetic  
15 fields in a circle around the accelerator pipe. Electrons and  
16 positrons at the end of the linear accelerator 10 enter the Beam  
17 Switch Yard (BSY) 34. Here the electrons are diverted in  
18 different directions by powerful dipole magnets 35 or connector  
19 magnets 35 and travel into storage rings 36, such as SPEAR, or  
20 into other experimental facilities or beamlines 38. To  
21 efficiently operate accelerator 10 operators constantly monitor  
22 all aspects of it.

23 **[0038]** The challenge to efficiently operate accelerator 10  
24 includes controlling temperature changes that cause the metal

1    accelerator structure to expand or contract. This expansion  
2    changes the frequency of the microwave resonance of the  
3    structure. Hence, the particle accelerator structure is  
4    preferably maintained at a steady temperature, throughout. The  
5    cooling system/process should be monitored to ensure all parts  
6    are working. Vacuum should also be maintained throughout the  
7    entire klystron waveguide, and accelerating structure. Any tiny  
8    vacuum leak interferes with accelerator function. The entire  
9    system is pumped out to 1/100,000,000,000 of atmospheric  
10   pressure. Further, the timing of the phase of each klystron  
11   must be correct, so that the entire structure, fed by numerous  
12   klystrons carries a traveling wave with no phase mismatches.  
13   Operators also monitor and focus the beam at many points along  
14   the accelerator. They use a variety of devices to monitor the  
15   beam such as strip beam position monitors (BPMs) and beam spot  
16   displays. Magnetic fields are typically used to focus the  
17   beams.

18   **[0039]**    After subatomic particles have been produced by  
19   colliding electrons and positrons, the subatomic particles must  
20   be tracked and identified. A particle can be fully identified  
21   when its *charge* and its *mass* are known.

22   **[0040]**    In principle the mass of a particle can be calculated  
23   from its momentum and *either* its speed or its energy. However,  
24   for a particle moving close to the speed of light any small

1   uncertainty in momentum or energy makes it difficult to  
2   determine its mass from these two, so it is necessary to measure  
3   speed as well.

4   **[0041]**     A multi-layer detector as shown in FIGURE 4 is used to  
5   identify particles. Each layer gives different information  
6   about the collision or interaction. Computer calculations based  
7   on the information from all the layers reconstruct the positions  
8   of particle tracks and identify the momentum, energy, and speed  
9   of as many as possible of the particles produced in the event.

10  **[0042]**     FIGURE 4 provides a cutaway schematic that shows all  
11  detector 50 elements installed inside a steel barrel and end  
12  caps. Complete detector may weigh as much as 4,000 tons and  
13  stands six stories tall. Innermost layer 52, the vertex  
14  detector, provides the most accurate information on the position  
15  of the tracks following collisions. The next layer, drift  
16  chamber 54, detects the positions of charged particles at  
17  several points along the track. The curvature of the track in  
18  the magnetic field reveals the particle's momentum. The middle  
19  layer, Cerenkov detector 56, measures particle velocity. The  
20  next layer, liquid argon calorimeter 58, stops most of the  
21  particles and measures their energy. This is the first layer  
22  that records neutral particles.

1   **[0043]**     A large magnetic coil 60 separates the calorimeter and  
2   the outermost layer 62. The outermost layer comprises magnet  
3   iron and warm iron calorimeter used to detect muons.

4   **[0044]**     The carefully controlled collisions within SLAC allow  
5   physicist to determine the fundamental (smallest) building  
6   blocks from which all matter is made and the interactions  
7   between the fundamental building blocks that govern how they  
8   combine and decay.

9   **[0045]**     The deployment of control solutions at SLAC further  
10  requires the development of device drivers that enable the  
11  adaptive control strategy with a nonlinear model predictive  
12  control technology to communicate to the distributed controls  
13  system (DCS) at SLAC and the installation of the adaptive  
14  control strategy with a nonlinear model predictive control  
15  technology at SLAC. The distributed control system at SLAC is  
16  also known as EPICS (Experimental Physics Industrial Control  
17  System).

18  **[0046]**     EPICS includes a set of software tools and  
19  applications which provide a software infrastructure with which  
20  to operate devices within the particle accelerators such as  
21  connector or quadrapole magnets or other like devices used to  
22  influence particle trajectories. EPICS represents in this  
23  embodiment a distributed control system comprising numerous  
24  computers, networked together to allow communication between

1 them and to provide control and feedback of the various parts of  
2 the device from a central room, or remotely over a network such  
3 as the internet.

4 **[0047]** Client/Server and Publish/Subscribe techniques allow  
5 communications between the various computers. These computers  
6 (Input/Output Controllers or IOCs) perform real-world I/O and  
7 local control tasks, and publish information to clients using  
8 network protocols that allow high bandwidth, soft real-time  
9 networking applications.

10 **[0048]** Such a distributed control system may be used  
11 extensively within the accelerator itself as well as by many of  
12 the experimental beamlines of SLAC. Numerous IOCs directly or  
13 indirectly control almost every aspect of the machine operation  
14 such as particle trajectories and environments, while  
15 workstations or servers in the control room provide higher-level  
16 control and operator interfaces to the systems/processes,  
17 perform data logging, archiving and analysis. Many IOCs can  
18 cause the accelerator to dump the beam when errors occur. In  
19 some cases a wrong output could damage equipment costing many  
20 thousands of dollars and days or even weeks to repair.

21 **[0049]** Architecturally, EPICS embodies the 'standard model'  
22 of distributed control system design. The most basic feature  
23 being that EPICS is fully distributed. Thus, EPICS requires no  
24 central device or software entity at any layer. This achieves



1 the goals of easy scalability, or robustness (no single point of  
2 failure).

3 **[0050]** EPICS comprises three physical layers as shown in  
4 FIGURE 5, and five software layers, as shown in FIGURE 6. The  
5 physical front-end layer is as the 'Input/Output Controller'  
6 (IOC) 70. Physical back-end layer 72 is implemented on popular  
7 workstations running Unix, or on PC hardware running Windows NT  
8 or Linux. Layers 70 and 72 are connected by network layer 74,  
9 which is any combination of media (such as Ethernet, FDDI, ATM)  
10 and repeaters and bridges supporting the TCP/IP Internet  
11 protocol and some form of broadcast or multicast.

12 **[0051]** The software layers utilize the 'client-server'  
13 paradigm. Client layer 76 usually runs in backend or  
14 workstation physical layer 72 and represents the top software  
15 layer. Typical generic clients are operator control screens,  
16 alarm panels, and data archive/retrieval tools. These are all  
17 configured with simple text files or point-and-click drawing  
18 editors.

19 **[0052]** The second software layer that connects all clients 76  
20 with all servers 78 is called 'channel access' (CA) 80. Channel  
21 access 80 forms the 'backbone' of EPICS and hides the details of  
22 the TCP/IP network from both clients 76 and servers 78. CA 80  
23 also creates a very solid 'firewall' of independence between all

1 clients and server code, so they can run on different  
2 processors. CA mediates different data representations.

3 **[0053]** The third software layer is the server layer 78. The  
4 fundamental server is the channel access server that runs on the  
5 target CPU embedded in every IOC. It insulates all clients from  
6 database layer 82. Server layer 78 cooperates with all channel  
7 access clients 76 to implement callback and synchronization  
8 mechanisms. Note that although clients 76 are typically  
9 independent host programs that call channel access 80 routines  
10 through a shared library, the channel access server is a unique  
11 distributed control task of the network nodes.

12 **[0054]** Database layer 82, is at the heart of the distributed  
13 control system. Using a host tool, the database is described in  
14 terms of function-block objects called 'records'. Record types  
15 exist for performing such chores as analog input and output;  
16 binary input and output; building histograms; storing waveforms;  
17 moving motors; performing calculations; implementing PID loops;  
18 emulating PALs, driving timing hardware; and other tasks.

19 Records that deal with physical sensors provide a wide variety  
20 of scaling laws; allowing smoothing; provide for simulation; and  
21 accept independent hysteresis parameters for display, alarm, and  
22 archive needs.

23 **[0055]** Record activity is initiated in several ways: from  
24 I/O hardware interrupts; from software 'events' generated by

1 clients 76 such as the Sequencer; when fields are changed from a  
2 'put'; or using a variety of periodic scan rates. Records  
3 support a great variety of data linkage and flow control, such  
4 as sequential, parallel, and conditional. Data can flow from  
5 the hardware level up, or from the software level down. Records  
6 validate data passed through from hardware and other records as  
7 well as on internal criteria, and can initiate alarms for un-  
8 initialized, invalid, or out-of-tolerance conditions. Although  
9 all record parameters are generated with a configuration tool on  
10 a workstation, most may be dynamically updated by channel access  
11 clients, but with full data independence. The fifth, bottom of  
12 layer of software is the device driver layer 84 for individual  
13 devices.

14 **[0056]** This distributed control system provides implements  
15 the 'standard model' paradigm. This control systems allows  
16 modularity, scalability, robustness, and high speed in hardware  
17 and software, yet remain largely vendor and hardware-  
18 independent.

19 **[0057]** The present invention provides a system and method of  
20 controlling particle collisions. To achieve this, specific  
21 algorithms have been developed that model and control the  
22 numerous variable associated with the linear accelerator at  
23 SLAC. Although the magnetic fields and their control have been  
24 specifically discussed here, it should be noted that these

1 algorithms may be applied to any variable associated with these  
2 structures. Further, it should be noted that this methodology  
3 has application beyond the control of particle accelerators.

4 **[0058]** The development of parametric nonlinear models with  
5 potentially varying parameters contributes to the design of  
6 successful control strategies for highly nonlinear dynamic  
7 control problems. The activities associated with the present  
8 invention are divided into two categories. The first category  
9 includes all the activities involved in developing the  
10 algorithms enabling the use of parameter varying nonlinear  
11 models within nonlinear model predictive control technology  
12 embodied in one implementation as Process Perfecter®. The  
13 second category includes all the activities involved in  
14 facilitating the deployment of the said controller.

15 **[0059]** The present invention treats all the variables upon  
16 which the current values of the varying parameters depend as  
17 inputs to the neural network model. This is illustrated in  
18 FIGURE 7. A separate NN maps input variables 93 to the  
19 varying parameters 95. At runtime, the values of the current  
20 input variables feed into NN 91 and the correct current varying  
21 parameter values are produced as the NN model outputs. The  
22 parameters in parametric model 97 are then updated to take on  
23 these values. Thus, the NN and the parametric models are  
24 connected in series. The combined model will then have correct

1 parameter values regardless of the operation region in which the  
2 system/process is operating.

3 **[0060]** The NN (its weights and biases) is trained as follows.  
4 The neural network is trained in the context of Figure 7. The  
5 inputs to the combined model are the process variable inputs 93,  
6 the outputs of the combined model are the process variable  
7 outputs 99. Any method used to train a NN as known to those  
8 skilled in the art may be used to train the NN in this combined  
9 structure. Any gradient method (including back propagation or  
10 any gradient-based nonlinear programming (NLP) method, such as a  
11 Sequential Quadratic Programming (SQP), a Generalized Reduced  
12 Gradient (GRG) or other like method known to those skilled in  
13 the art) requires that the parametric model 97 be  
14 differentiable, while non-gradient methods do not impose this  
15 restriction.

16 **[0061]** Any gradient-based method requires the gradients of  
17 the error with respect to the weights and biases. These  
18 gradients can be readily obtained (assuming the models are  
19 differentiable) in either numerical or analytical derivatives.  
20 Numerical approximations to the derivatives are computed by  
21 making small changes to a weight/bias, observing the resulting  
22 process variable output, and then making one or more additional  
23 different and small change to the weight/bias, and again

1 observing the FP output. An appropriate formula for first  
2 derivative approximation is then used.

3 **[0062]** The gradient of the error with respect to any of the  
4 NN weights and biases can be computed via the chain rule for  
5 derivatives. Hence, gradient-based methods require the  
6 Parametric model 97 to be differentiable.

7 **[0063]** The NN is trained without explicit targets for its own  
8 outputs. The NN outputs are in the same position in the  
9 combined model as are the hidden units in a NN - the errors for  
10 the NN outputs originate from the targets at the process  
11 variable output 99 level.

12 **[0064]** Any non-gradient method ordinarily requires that the  
13 process outputs 99 be computed as the first step, of and the  
14 chosen method's own evaluation of the goodness of the current  
15 state of the combined model is determined readily from any of  
16 the needed values within the combined model. Typically, non-  
17 gradient methods use error as the measure of goodness.

18 **[0065]** The present invention may utilize any parametric model  
19 structure whatsoever for the FP model block 97: steady state  
20 models, including those represented by open and by closed  
21 equations, and including whether or not the FP outputs are all  
22 separable to the left hand side of the equations or not, and  
23 whether or not all of the FP outputs are measured, as well as

1 dynamic models, including IIR, FIR, difference equation, and  
2 differential equation models.

3 **[0066]** The methodology by which variation in process dynamics  
4 over different operation regimes is incorporated in the  
5 nonlinear model predictive control solution is described below.  
6 This invention's handling of systems with variable dynamics  
7 provides a commercially viable solution to a long-standing  
8 demand for robust adaptive control strategies in industry.

9 **[0067]** Significant applications exist in which dynamic  
10 behavior at the process varies considerably over the expected  
11 operation region. Examples range from polystyrene process and  
12 reactors with significant variation in the residence time, to  
13 acoustic systems/processes with temperature dependent acoustic  
14 properties, and supersonic airplanes operating over a wide range  
15 of mach numbers. As previously described, one embodiment of the  
16 present invention focuses on the application to the control of a  
17 linear accelerator. However, the present invention need not be  
18 so limited.

19 **[0068]** Relevant information regarding accurate description of  
20 the system/process dynamics under these circumstances can be  
21 found from a variety of resources. They include first-  
22 principles equations capturing functional dependency of dynamic  
23 parameters on input/output variables, operator knowledge, and

1 empirical data rich enough to adequately represent changes in  
2 system/process dynamics.

3 **[0069]** The absence of a systematic way for handling varying  
4 process dynamics forces application engineers to devote  
5 significant energy and time so that the variations in process  
6 dynamics does not result in serious degradation of the  
7 controller performance. The present invention extends the  
8 existing formulations such that variations in process dynamics  
9 can be properly considered. This may result in improved  
10 input/output controller (IOC) performance as well as expanded  
11 operating conditions. The derivation of the proposed algorithm  
12 is based on the following general representation for the  
13 dynamics of the process as a nonlinear, possibly time-varying  
14 difference equation:

$$15 \quad Y_K = F(u_K, u_{K-1}, \dots, u_{K-M}, y_{K-1}, \dots, y_{K-N}) \quad (7)$$

16 where  $u_k$  is the vector of input variables affecting the  
17 process (i.e., both manipulated and disturbance variable  
18 inputs),  $y_k$  is the vector of measured outputs, and  $F$  is a  
19 potentially time-varying nonlinear vector function.

20 In one embodiment, the present invention proposes the following  
21 perturbation model to locally approximate Equation (5):

$$22 \quad \delta y_k = \sum_{i=1}^N \alpha(u_k, u_{k-1}, \dots, u_{k-M}, y_{k-1}, \dots, y_{k-N}) \delta y_{k-1} + \sum_{i=1}^M \beta(u_k, u_{k-1}, \dots, u_{k-M}, y_{k-1}, \dots, y_{k-N}) \delta y_{k-1} \quad (6)$$

23 where the coefficients  $\alpha(\cdot)$  and  $\beta(\cdot)$  can be defined as:



$$\alpha(u_k, u_{k-1}, \dots, u_{k-M}, y_{k-1}, \dots, y_{k-N}) = \frac{\partial F}{\partial y_{k-1}} \quad (7)$$

and

$$\beta(u_k, u_{k-1}, \dots, u_{k-M}, y_{k-1}, \dots, y_{k-N}) = \frac{\partial F}{\partial u_{k-1}} \quad (8)$$

are functions of present and past inputs/outputs of the system. The methodology presented in this invention is applicable for higher order local approximations of the nonlinear function  $F$ . Also, as mentioned earlier, for a given state-space representation of a nonlinear parameter-varying system, an equivalent input/output model with the representation of Equation (5) can be constructed in a variety of ways known to experts in the field. Hence, the methodology presented here encompasses systems described in state-space as well. The approximation strategy captured by Figure 7 is directly applicable to any functional mapping from an input space to output space, and hence the approach in this invention is directly applicable to state space description of the linear processes with varying dynamics.

**[0070]** This algorithm encompasses case where non-linearity in the parameters of the dynamic model (in addition to the gain) is explicitly represented.

1 [0071] The information regarding variation in dynamic  
2 parameters of the process can be directly incorporated in the  
3 controller design regardless of the source of the information  
4 about varying parameters.

5 [0072] The present invention may be applied whether complete  
6 or partial knowledge of the dynamic parameters is available.  
7 When full information regarding process dynamic parameters is  
8 available,  $\alpha(u_k, u_{k-1}, \dots, u_{k-M}, y_{k-1}, \dots, y_{k-N}) = \frac{\partial F}{\partial y_{k-1}}$  and  $\beta(u_k, u_{k-1}, \dots, u_{k-M}, y_{k-1}, \dots, y_{k-N}) = \frac{\partial F}{\partial u_{k-1}}$

9 's in Equations. (6-8) are explicitly defined by the user.  
10 However, in the case of partial information, only some of the  
11 parameters are explicitly defined and the rest are found via an  
12 identification algorithm from empirical data.

13 [0073] Where second order models are used to describe the  
14 process, users most often provide information in terms of gains,  
15 time constants, damping factors, natural frequencies, and delays  
16 in the continuous time domain. The translation of these  
17 quantities to coefficients in a difference equation of the type  
18 shown in Equation (6) is straightforward and is given here for  
19 clarity:

20 For a system/process described as  $\frac{k}{(T\delta + 1)}$ , the difference  
21 equation based on ZOH discretization is:

$$\delta y_k = \left( e^{-\frac{T}{\tau}} \right) \delta y_{k-1} + k \left( 1 - e^{-\frac{T}{\tau}} \right) \delta u_{k-1} \quad (9)$$

For an over-damped system/process described as  $\frac{k(\tau_{lead}\zeta + 1)}{(\tau_1\zeta + 1)(\tau_2\zeta + 1)}$  the difference equation is:

$$\begin{aligned} \delta y_k = & \left( e^{-\frac{T}{\tau_1}} + e^{-\frac{T}{\tau_2}} \right) \delta y_{k-1} - \left( e^{-\left(\frac{T}{\tau_1} + \frac{T}{\tau_2}\right)} \right) \delta y_{k-2} \\ & + \left( A \left( 1 - e^{-\frac{T}{\tau_1}} \right) + B \left( 1 - e^{-\frac{T}{\tau_2}} \right) \right) \delta u_{k-1} \\ & - \left( A e^{-\frac{T}{\tau_2}} \left( 1 - e^{-\frac{T}{\tau_1}} \right) + B e^{-\frac{T}{\tau_1}} \left( 1 - e^{-\frac{T}{\tau_2}} \right) \right) \delta u_{k-2} \end{aligned} \quad (10)$$

where

$$A = k \frac{\tau_1 - \tau_3}{\tau_1 - \tau_2}$$

and

$$B = k \frac{\tau_3 - \tau_2}{\tau_1 - \tau_2}.$$

For a system/process described as  $\frac{k(\tau_{lead}\zeta + 1)}{(\tau\zeta + 1)^2}$ , the difference equation is:

$$= \left( 2e^{-\frac{T}{\tau}} \right) \delta y_{k-1} - \left( e^{-2\frac{T}{\tau}} \right) \delta y_{k-2}$$

$$\begin{aligned}
& + \left( k - ke^{-\frac{T}{\tau}} \left( 1 + \frac{T}{\tau} - \frac{\tau_{lead} T}{\tau^2} \right) \right) \delta u_{k-1} \\
& + \left( ke^{-\frac{2T}{\tau}} - ke^{-\frac{T}{\tau}} \left( 1 - \frac{T}{\tau} - \frac{\tau_{lead} T}{\tau^2} \right) \right) \delta u_{k-2}
\end{aligned} \tag{11}$$

For an under-damped system/process described as  $\frac{k(\tau_{lead}\delta + 1)}{\delta^2 + 2\frac{\zeta}{\tau}\delta + \frac{1}{\tau^2}}$  the

difference equation is:

$$\begin{aligned}
\delta y_k = & \left( 2e^{-\frac{\zeta T}{\tau}} \cos \left( \frac{\sqrt{1-\zeta^2}}{\tau} T \right) \right) \delta y_{k-1} - \left( e^{-\frac{2\zeta T}{\tau}} \right) \delta y_{k-2} \\
& + \left( \frac{G}{B} e^{-\frac{\zeta T}{\tau}} \sin \left( \frac{\sqrt{1-\zeta^2}}{\tau} T \right) + kA_1 \right) \delta u_{k-1} \\
& + \left( -\frac{G}{B} e^{-\frac{\zeta T}{\tau}} \sin \left( \frac{\sqrt{1-\zeta^2}}{\tau} T \right) + kA_2 \right) \delta u_{k-2}
\end{aligned} \tag{12}$$

where

$$G = \frac{k\tau_{lead}}{\tau^2}$$

$$B = \frac{\sqrt{1-\zeta^2}}{\tau}$$

$$A_1 = 1 - e^{-\frac{\zeta T}{\tau}} \cos \left( \frac{\sqrt{1-\zeta^2}}{\tau} T \right) - \frac{\zeta}{\sqrt{1-\zeta^2}} e^{-\frac{\zeta T}{\tau}} \sin \left( \frac{\sqrt{1-\zeta^2}}{\tau} T \right),$$

and

$$A_2 = e^{-\frac{2\zeta}{\tau}T} - e^{-\frac{\zeta}{\tau}T} \cos\left(\frac{\sqrt{1-\zeta^2}}{\tau}T\right) + \frac{\zeta}{\sqrt{1-\zeta^2}} e^{-\frac{\zeta}{\tau}T} \sin\left(\frac{\sqrt{1-\zeta^2}}{\tau}T\right).$$

[0074] The present invention accommodates user information whether there is an explicit functional description for the parameters of the dynamic model, or an empirical model is built to describe the variation, or just a tabular description of the variations of the parameters versus input/output values.

[0075] During optimization, the solver may access the available description for the variation of each parameter in order to generate relevant values of the parameter given the current and past values of the input(s)/output(s). Numerical efficiency of the computations may require approximations to the expressed functional variation of the parameters.

[0076] The present invention preserves the consistency of the steady-state neural network models and the dynamic model with varying dynamic parameters.

[0077] Using an approximation to the full dynamic model can simplify the implementation and speed up the execution frequency of the controller. The following details one such an approximation strategy. This invention, however, applies regardless of the approximation strategy that is adopted. Any approximation strategy known to those skilled in the art is therefore incorporated by reference in this disclosure.

1   **[0078]**     The models may be updated when (a) changes in control  
2   problem setup occur (for example setpoint changes occur), or (b)  
3   when users specifically ask for a model update, or (c) when a  
4   certain number of control steps, defined by the users, are  
5   executed, or (d) an event triggers the update of the models.

6   **[0079]**     Assuming that  $(u_{init}, y_{init})$  is the current operating  
7   point of the system/process, and  $y_{final}$ , is the desired value of  
8   the output at the end of the control horizon, the present  
9   invention utilizes the steady state optimizer to obtain  $u_{final}$   
10  that corresponds to the desired output at the end of the control  
11  horizon.

12  **[0080]**     The dynamic difference equation is formed at the  
13  initial and final points, by constructing the parameters of the  
14  dynamic model given the initial and final operation points,  
15   $(u_{init}, y_{init})$  and  $(u_{final}, y_{final})$  respectively. Note that the  
16  functional dependency of the parameters of the dynamic model on  
17  the input/output values is well-defined (for example, user-  
18  defined, tabular, or an empirical model such as a NN.).

19  **[0081]**     To approximate the difference equation during  
20  process's transition from initial operation point to its final  
21  operation point, one possibility is to vary the parameters  
22  affinely between their two terminal values. This choice is for  
23  ease of computation, and the application of any other  
24  approximation for the parameter values in between (including but

1 not limited to higher order polynomials, sigmoid-type function,  
 2 and tangent hyperbolic function) as is known to those skilled in  
 3 the art may also be employed. To highlight the generality of  
 4 the approach in this invention, the present invention may follow  
 5 affine approximation of the functional dependency of parameters  
 6 on input/output values is described here. Assume that  $p$  is a  
 7 dynamic parameter of the system/process such as time constant,  
 8 gain, damping, etc. Parameter  $p$  is a component of the FPM  
 9 parameters 95 in Figure 7. Also assume that  $p = f(u_k, u_{k-1}, \dots, u_{k-M},$   
 10  $y_{k-1}, \dots, y_{k-N})$ , where  $f$  is an appropriate mapping. Note that with  
 11 the assumption of steady state behavior at the two ends of the  
 12 transition  $u_k = u_{k-1} = \dots = u_{k-M}$  and  $y_{k-1} = y_{k-2} = \dots = y_{k-N}$ . An affine  
 13 approximation for this parameter can be defined as follows:

$$14 \quad p(u_k, u_{k-1}, y_{k-1}, y_{k-2}) = p(u_{init}, y_{init}) + p_u \left( \frac{\partial p}{\partial u} \right)_{init} (u_k - u_{init}) + p_y \left( \frac{\partial p}{\partial y} \right)_{init} (y_k - y_{init}) \quad (13)$$

15 where for simplicity  $M=N=2$  is assumed.

16 When state space description of the process is available  $p$  may  
 17 be a function of state as well. The methodology is applicable  
 18 regardless of the functional dependency of  $p$ .

19 **[0082]** Note that the coefficients  $p_u$  and  $p_y$  are approximation  
 20 factors and must be defined such that  $p(u_{final}, y_{final}) = f(u_{final},$   
 21  $y_{final})$ , where the following substitutions are done for brevity:  
 22  $u_k = u_{k-1} = \dots = u_{k-M} = u_{final}$  and  $y_{k-1} = \dots = y_{k-N} = y_{final}$ . The constraint on the  
 23 final gain is not enough to uniquely define both  $p_u$  and  $p_y$ , This

1 present invention covers all possible selections for  $p_u$  and  $p_y$ .  
 2 One possible option with appropriate scaling, and  
 3 proportionality concerns is the following:

$$4 \quad p_u = \left( \frac{p_{final} - p_{init}}{u_{final} - u_{init}} \right) \frac{1}{\frac{\partial p}{\partial u} + \varepsilon \frac{\partial p}{\partial y}} \quad (14)$$

$$5 \quad p_y = \left( \frac{p_{final} - p_{init}}{y_{final} - y_{init}} \right) \frac{\varepsilon}{\frac{\partial p}{\partial u} + \varepsilon \frac{\partial p}{\partial y}} \quad (15)$$

6 where  $0 \leq \varepsilon \leq 1$  is a parameter provided by the user to  
 7 determine how the contributions from variations in  $u_k$  and  $y_k$   
 8 must be weighted. By default  $\varepsilon$  is 1.

9 **[0083]** The quantities  $\frac{\partial p}{\partial u}$  and  $\frac{\partial p}{\partial y}$  can be provided in  
 10 analytical forms by the user. In the absence of the analytical  
 11 expressions for these quantities, they can be approximated. One  
 12 possible approximation is  $\left( \frac{p_{final} - p_{init}}{u_{final} - u_{init}} \right)$  and  $\left( \frac{p_{final} - p_{init}}{y_{final} - y_{init}} \right)$   
 13 respectively.

14 **[0084]** To maintain the coherency of the user-provided  
 15 information regarding dynamic behavior of the process, and the  
 16 information captured by a steady-state neural network based on  
 17 empirical data, an additional level of gain scheduling is  
 18 considered in this invention. The methodology describing this  
 19 gain scheduling is described in detail. .



[0095b] One possible approach for maintaining the consistency of the static nonlinear gain information with the dynamic model is described below. This invention however need not be limited to the approach described here.

1. The difference equation of the type described by Equation (6) is constructed. For example, the variable dynamics information on  $\tau$ ,  $\zeta$ , lead time, etc. at the initial and final points will be translated into difference model in Equation (6) using Equations (9)-(12).

2. The overall gain at the initial and final point is designed to match that of the steady state neural network, or that of the externally-provided variable dynamics gain information:

(a) From the static neural network the gains at each operation point, i.e.  $(g_i^{ss} = \frac{dy}{du})_{(u_{init}, y_{init})}$ , and  $(g_f^{ss} = \frac{dy}{du})_{(u_{final}, y_{final})}$ , are extracted. User can also define the gain to be a varying parameter.

(b) For simplicity of the presentation, a second order difference equation is considered here:

$$\begin{aligned} \delta y_k = & -a_1(.) \delta y_{k-1} - a_2(.) \delta y_{k-2} \\ & + \nu_1 \delta u_{k-1-\Delta} + \nu_2 \delta u_{k-2-\Delta} \\ & + \omega_1 (u_{k-1} - u_{init}) \delta u_{k-1-\Delta} + \omega_2 (u_{k-2} - u_{init}) \delta u_{k-2-\Delta} \end{aligned} \quad (12)$$

where  $a_1(.)$  and  $a_2(.)$  can be constructed as follows:

$$a_1(.) = \left( a_1^i + (a_1^f - a_1^i) \frac{\bar{u}_{k-1} - u_{init}}{u_{final} - u_{init}} \right)$$

$$a_2(.) = \left( a_2^i + (a_2^f - a_2^i) \frac{\bar{u}_{k-2} - u_{init}}{u_{final} - u_{init}} \right)$$

where  $a_1^i, a_1^f, a_2^i, a_2^f, b_1^i, b_1^f, b_2^i, b_2^f$  are determined using Equations (9)-(12).

$\bar{u}_{k-1}$  and  $\bar{u}_{k-2}$  can be defined (but need not be limited to) the following:

$$\bar{u}_k = u_i + \frac{1}{2} (u_f - u_i) \left( 1 + \frac{e^{\kappa \frac{u_k - u_m}{u_r}} - e^{-\kappa \frac{u_k - u_m}{u_r}}}{e^{\kappa \frac{u_k - u_m}{u_r}} + e^{-\kappa \frac{u_k - u_m}{u_r}}} \right)$$

where  $u_m = \frac{u_f + u_i}{2}$ ,  $u_r = \|u_f - u_i\|$  and  $\kappa$  is a parameter that controls how the transition from  $u_i$  to  $u_f$  will occur. If no varying parameter exists, then the initial and final values for these parameters will be the same.

(c) Parameters  $\nu_1, \nu_2, \omega_1, \omega_2$  must then be defined such that the steady state gain of the dynamic system matches those extracted from the neural network at both sides of the transition region (or with the externally-provided gain information that is a part of variable dynamics description). One possible selection for the parameters is (but need not be limited to) the following:

$$\nu_1 = b_1^i \left( \frac{1 + a_1^i + a_2^i}{b_1^i + b_2^i} \right) g_{ss}^i$$

$$\nu_2 = b_2^i \left( \frac{1 + a_1^i + a_2^i}{b_1^i + b_2^i} \right) g_{ss}^i$$

(d) A possible selection for  $\omega_1$  and  $\omega_2$  parameters is (but

need not be limited to) the following:

$$\omega_1 = \left( \frac{b_1^f}{b_1^f + b_2^f} \right) \left( \frac{1 + a_1^f + a_2^f}{u_{final} - u_{init}} \right) g_{ss}^f - \frac{\nu_1}{u_{final} - u_{init}}$$

$$\omega_2 = \left( \frac{b_2^f}{b_1^f + b_2^f} \right) \left( \frac{1 + a_1^f + a_2^f}{u_{final} - u_{init}} \right) g_{ss}^f - \frac{\nu_2}{u_{final} - u_{init}}$$

**[0085]** The present invention in one embodiment may be applied towards modeling and control at the linear accelerator at SLAC. The present invention further includes the development device drivers that enable communication between the Data Interface of the present invention (DI) and SLAC's EPICS that talks to the lower level Distributed Control System at SLAC.

**[0086]** Any communication between the hardware and a control system such as the one at SLAC is done through SLAC's EPICS system, and therefore, the present invention includes a reliable interface between the hardware and the control system.

**[0087]** The results from the modeling effort on the collected data on SPEAR II are summarized in FIGURES 8, 9, and 10. A quick look at the relevant data captured in the course of one experiment where three manipulated variables were intentionally moved in the course of the experiments: two corrector magnets and one quadrapole magnet. The reading of Beam Position Monitors is recorded as the controlled variables or output of this experiment.

1   **[0088]**     Screen capture 100 of the input/output variables from  
2   the test data is provided in FIGURE 8. Note that the x and y  
3   reading of one of the BPMs are chosen as the MVs are the ones  
4   mentioned earlier, the tag name for which is clearly indicated  
5   in the screen capture. FIGURE 8 evidences the clear correlation  
6   between the MVs with the BPM. Another screen analytic is  
7   provided in FIGURE 9 gives a better screenshot 110 of the  
8   variation in variables.

9   **[0089]**     FIGURE 10 provides yet another screen shot 120 where  
10   the dots 122 are actual data points. A model of the nonlinear  
11   input/output relationship was constructed using Pavilion's  
12   Perfecter®. Due to simultaneous variation in manipulated  
13   variables, the identification is rather difficult. Data is  
14   manipulated (by cutting certain regions of data) to make sure  
15   that the maximum accuracy in the identification of the  
16   input/output behavior is captured.

17   **[0090]**     FIGURE 10 displays one such input/output relationship  
18   for the SPEAR Equipment at SLAC. This figure clearly shows the  
19   nonlinear input/output relationship in the above-mentioned  
20   model.

21   **[0091]**     The present invention's capability in the design of  
22   new adaptive control algorithms, identification of processes  
23   with varying dynamics is clearly demonstrated. Further

1 development efforts will improve the developed algorithms to a  
2 commercial quality code base.

3 **[0092]** In summary, the present invention provides a method  
4 for controlling nonlinear control problems in operating  
5 processes like a particle accelerator. The invention utilizes  
6 modeling tools to identify variable input and controlled  
7 variables associated with the process, wherein at least one  
8 variable input is a manipulated variable input. The modeling  
9 tools are further operable to determine relationships between  
10 the variable inputs and controlled variables. A control system  
11 that provides inputs to and acts on inputs from the modeling  
12 tools tunes one or more manipulated variables to achieve a  
13 desired controlled variable, which in the case of a particle  
14 accelerator may be realized as a more efficient collision.

15 **[0093]** FIGURE 12 provides another illustration of the  
16 relationship of the process 200 and the controller 202 and more  
17 importantly the relationship of the models 204, 206 and 208  
18 within the controller 202 to the control of the process 200. A  
19 typical process has a variety of variable inputs  $u(t)$  some of  
20 these variables may be manipulated variable inputs 210 and some  
21 may be measured disturbance variables 212 and some may be  
22 unmeasured disturbance variables 214. A process 200 also  
23 typically has a plurality of variable outputs. Some are  
24 measurable and some are not. Some may be measurable in real-

1 time 220 and some may not 222. Typically, a control systems  
2 objective is to control one of these variable outputs this  
3 variable is can be called the control variable or controlled  
4 variable. Additionally, to the controller the variable outputs  
5 may be considered one of the variable inputs to the controller  
6 or controller variable inputs 223. Typically but not  
7 necessarily, a control system uses a distributed control system  
8 (DCS) 230 to manage the interactions between the controller 202  
9 and the process 200 - as illustrated in the embodiment in FIGURE  
10 12. In the embodiment shown the controller includes a steady  
11 state model 204 which can be a parameterized physical model of  
12 the process. This model can receive external input 205  
13 comprised of the desired controlled variable value. This may or  
14 may not come from the operator or user (not shown) of the  
15 process/control system 202. Additionally the embodiment  
16 illustrates a steady state parameter model 206 that maps the  
17 variable inputs  $u$  to the variable output(s)  $y$  in the steady  
18 state model. Further, the embodiment illustrates a variable  
19 dynamics model which maps the variable inputs  $u$  to the  
20 parameters  $p$  of the parameterized physical model (steady state  
21 model) of the process. In one embodiment of the invention  
22 empirical modeling tools in this case NNs are used for the  
23 Steady State parameter model and the variable dynamics parameter  
24 models. Based on input received from the process these models

1 provide information to the dynamic controller 232 which can be  
2 optimized by the optimizer 234. The Optimizer is capable of  
3 receiving optimizer constraints 236 which may possibly receive  
4 partial or possibly total modification from an external source  
5 238, which may or may not be the operator or user (not shown) of  
6 the process 200 or control system 202. Inputs 205 and 208 may  
7 come from sources other than the operator or user of the control  
8 system 202. The dynamic controller 232 provides the information  
9 to the DCS 230 which sends provides setpoints for the  
10 manipulated variable inputs 240 which is the output of the  
11 controller 240.

12 **[0094]** Although the particle accelerator example is described  
13 in great detail, the inventive modeling and control system  
14 described herein can be equally applied to other operating  
15 processes with comparable behavioral characteristics. For  
16 example, temperature control in a manufacturing plant such as a  
17 polymer manufacturing plant, or load-frequency control in a  
18 power grid would all benefit from the present inventive control  
19 system.

20 **[0095]** Although the present invention is described in detail,  
21 it should be understood that various changes, substitutions and  
22 alterations can be made hereto without departing from the spirit  
23 and scope of the invention as described by the appended claims.